BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images, please do not report the images to the Image Problem Mailbox.

(19) RÉPUBLIQUE FRANÇAISE

INSTITUT NATIONAL
DE LA PROPRIÉTÉ INDUSTRIELLE

PARIS

(11) N° de publication : **2 693 008**
(à n'utiliser que pour les commandes de reproduction)

(21) N° d'enregistrement national : **92 08048**

(51) Int Cl[5] : G 06 F 12/08

(12) DEMANDE DE BREVET D'INVENTION A1

(22) **Date de dépôt** : 30.06.92.

(30) **Priorité** :

(43) **Date de la mise à disposition du public de la demande** : 31.12.93 Bulletin 93/52.

(56) **Liste des documents cités dans le rapport de recherche préliminaire** : *Se reporter à la fin du présent fascicule.*

(60) **Références à d'autres documents nationaux apparentés :**

(71) **Demandeur(s)** : *Société anonyme dite: GEMPLUS CARD INTERNATIONAL* — FR.

(72) **Inventeur(s)** : Sourenian Paul.

(73) **Titulaire(s) :**

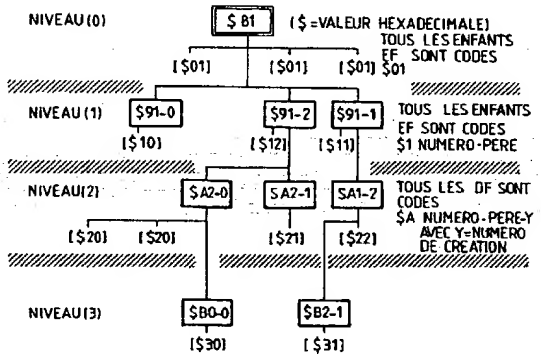
(74) **Mandataire** : Cabinet Ballot-Schmit.

(54) Procédé de reconnaissance des fichiers dans une mémoire, notamment une mémoire pour carte à circuit intégré.

(57) L'invention concerne les procédés qui permettent de structurer la mémoire d'un système informatique selon une arborescence hiérarchique.

Elle consiste à placer en tête de la mémoire une zone système comportant le programme d'exploitation (101) puis ensuite un répertoire principal (102) suivi de répertoires secondaires (104, 108) et de fichiers élémentaires (106, 109). Ces répertoires secondaires et ces fichiers élémentaires sont situés les uns après les autres au fur et à mesure de leur création mais en respectant au niveau des répertoires secondaires l'ordre hiérarchique. Un codage particulier permet à partir de la sélection au niveau du système d'exploitation d'une application particulière correspondant à un ensemble de répertoires secondaires et de fichiers élémentaires, de retrouver rapidement ceux-ci sans risque de s'égarer dans les autres applications.

Elle permet de sélectionner facilement les applications tout en consommant peu d'espace pour le codage.



FR 2 693 008 - A1

PROCEDE DE RECONNAISSANCE DES FICHIERS DANS UNE MEMOIRE, NOTAMMENT UNE MEMOIRE POUR CARTE A CIRCUIT INTEGRE.

La présente invention se rapporte aux procédés qui permettent de reconnaître de manière facile les fichiers inscrits dans une mémoire, tout en consommant peu de place dans cette mémoire. Elle s'applique plus particulièrement aux mémoires à semi-conducteurs intégrées dans des cartes à circuits intégrés, plus connues sous le nom de "cartes à puce ".

Il est maintenant courant d'utiliser des cartes normalisées, tant au point de vue dimensions que connecteur, dans lesquelles sont insérées des circuits intégrés qui permettent différentes opérations. La plus connue est la carte à puce utilisée pour les cabines téléphoniques, qui comporte une simple mémoire, laquelle est consommée au fur et à mesure de l'utilisation. Des applications beaucoup plus élaborées sont déjà en service, par exemple celles concernant le décryptage des signaux de télévision cryptés. Ces applications élaborées nécessitent généralement l'usage d'un microprocesseur faisant partie de la puce, ou éventuellement des puces insérée dans la carte. Avec un tel microprocesseur on peut alors étendre l'usage d'une carte unique à des applications différentes ou à des utilisateurs différents, voire même aux deux.

Les données nécessaires à ces applications multiples, ainsi que celles créées lors des utilisations, sont contenues dans une mémoire, laquelle est classiquement organisée selon des fichiers qui sont affectés aux diverses applications.

La gestion de cette mémoire implique d'avoir une structure permettant d'une part de repérer facilement

les fichiers, même s'ils sont physiquement éparpillés dans toute l'étendue de la mémoire et d'autre part de ne permettre lorsqu'une application est en cours d'exécution que l'accès aux fichiers qui lui sont réservés et uniquement à ceux-là. Pour cela, il est connu d'avoir en tête du fichier un ensemble de données appelé descripteur qui contient toutes les indications nécessaires aux repérages et aux autorisations et interdictions d'accès.

On connaît de telles organisations, qui sont souvent arborescentes et hiérarchiques et où les relations entre les différents fichiers sont fixées par une série de répertoires liés entre eux et aux fichiers finals par une relation du type "parents/enfants".

Une telle organisation n'est pas en soi très difficile à mettre en oeuvre, mais l'organisation des répertoires consomme beaucoup de place mémoire, et dès que la structure devient un peu complexe le volume utilisé par les descripteurs des différents répertoires et fichiers devient très important, ce qui est prohibitif dans une mémoire destinée à être placée dans une carte à puce.

Pour pallier cet inconvénient, l'invention propose procédé de reconnaissance des fichiers dans une mémoire, notamment une mémoire pour carte à circuit intégré, du type consistant à prévoir une organisation hiérarchique arborescente du type "parents-enfants" ayant en tête un répertoire principal MF suivi de répertoires secondaires DF et de fichiers élémentaires EF, principalement caractérisé en ce que l'on place dans la mémoire le répertoire principal en tête puis les répertoires secondaires et les fichiers élémentaires à la suite les uns des autres au fur et à mesure de leur création mais en respectant l'ordre hiérarchique, que l'on code le MF

et les DF avec un premier mot binaire comprenant un digit ou un bit indiquant qu'il s'agit du MF ou d'un DF et un ensemble d'autres bits indiquant le niveau hiérarchique du répertoire et un 2ème mot binaire indiquant le numéro du répertoire parent dans son niveau hiérarchique, et pour les DF un 3ème mot binaire permettant de coder le numéro de ce DF à l'intérieur de son niveau hiérarchique, et que l'on code en outre les fichiers élémentaires EF avec un premier mot binaire comportant un bit indiquant qu'il s'agit d'un EF, ce bit étant l'inverse du bit correspondant du MF ou des DF et les autres bits de ce premier mot binaire indiquant le niveau hiérarchique de l'EF, et un deuxième mot binaire identique au troisième mot binaire du DF dont l'EF est l'enfant.

D'autres particularités et avantages de l'invention apparaîtront clairement dans la description suivante, présentée à titre d'exemple non limitatif en regard des figures annexées qui représentent :

- La figure 1, un exemple d'organisation hiérarchique d'un ensemble de répertoires et de fichiers;
- la figure 2, le codage de l'ensemble de la figure 1 par le procédé selon l'invention; et
- la figure 3, un exemple de répartition de répertoires et de fichiers dans une mémoire par le procédé selon l'invention.

On a représenté sur la figure 1 une structure arborescente et hiérarchisée organisée selon une relation parents/enfants permettant de mettre en oeuvre le procédé selon l'invention.

Dans cet exemple les fichiers sont répartis sur quatre niveaux 0 à 3.

Dans le niveau 0 on trouve un répertoire principal

dit MF qui est unique, n'a pas de parent mais peut avoir des enfants du type DF ou EF comme définis ci-dessous.

Les autres éléments de cette organisation sont donc des répertoires secondaires appelés DF et des fichiers élémentaires appelés EF.

Les répertoires secondaires DF peuvent avoir comme parent, soit le répertoire principal MF soit un autre DF. Ils peuvent avoir des enfants, dont ils sont donc le parent, qui sont eux-mêmes soit d'autres répertoires secondaires DF, soit des fichiers élémentaires EF.

Les fichiers EF sont ceux qui contiennent les données proprement dites nécessaires à la mise en oeuvre des applications, et les répertoires, tant le principal MF que les secondaires DF, ne contiennent que les données permettant au système d'exploitation d'aboutir aux fichiers EF à utiliser, en suivant les chemins qui y aboutissent tout au long de l'arborescence.

Ainsi donc sur l'exemple de la figure 1, on a au niveau 0 le répertoire principal MF qui à trois fichiers EF0 comme enfants.

Au niveau 1 on trouve 3 répertoires secondaires DF0, DF1 et DF2 qui sont les enfants du MF. DF0 à 1 enfant unique EF0. DF1 à 2 enfants, un fichier EF1 (de niveau 1) et un répertoire secondaire DF2-1 du niveau 2. DF2 lui n'a pas de fichier élémentaire enfant au niveau 1, mais il a comme enfants 2 répertoires secondaires DF0-2 et DF1-2 au niveau 2.

Dans ce niveau 2, DF0-2 a deux fichiers EF0 comme enfants, et DF1-2 et DF2-1 ont respectivement les fichiers EF1 et EF2 comme enfants à ce niveau 2. Enfin au niveau 3, le DF0-2 du niveau 2 a un répertoire secondaire enfant DF0-2 qui a lui même un fichier enfant EF0. Toujours dans ce niveau 3 le répertoire secondaire DF2-1 du niveau 2 a un enfant répertoire secondaire

DF1-1 qui a lui même un enfant fichier EF1.

Comme on le voit sur cet exemple simple, tout le problème auquel répond l'invention est de coder de manière simple et efficace les répertoires et les fichiers pour que le système d'exploitation sache toujours où il en est.

Dans l'exemple décrit, selon l'invention le codage d'un DF s'effectue sur 12 bits divisés en 3 quartets de 4 bits, et les codages du MF et des EF s'effectuent sur 8 bits répartis en 2 quartets de 4 bits.

Considérons tout d'abord le codage du MF et des DF

Le 1er quartet comprend 4 bits dont le 1er (c'est à dire le bit de poids fort) est toujours un 1 pour indiquer qu'il s'agit d'un répertoire, qu'il soit principal ou secondaire. Les 3 autres bits peuvent prendre des valeurs allant de 000 à 111, ce qui permet de coder 8 niveaux hiérarchiques. Le 1er niveau hiérarchique codé 000 correspond au MF, et les 7 autres, codés de 001 à 111 correspondent aux différents DF situés en dessous du MF.

Le 2ème quartet permet de coder le numéro du DF parent du DF codé en question, qui est situé bien entendu dans le niveau supérieur. Comme ce quartet permet de coder de 0000 à 1111 en binaire, soit 0 à 15 en décimal ou 0 à F hexadécimal, ceci signifie qu'il n'y a que 16 DF parents possible dans chaque niveau supérieur et donc que ce système hiérarchique ne permet de définir pour chaque niveau que 16 DF.

Les DF ont un 3ème quartet qui permet de coder le numéro du DF dans un même niveau, pour différencier les 16 DF qui peuvent être présents dans un même niveau hiérarchique, comme on l'a vu ci-dessus. Là aussi ce quartet peut aller de 0 à F en hexadécimal.

On a représenté dans le tableau suivant les valeurs

possible pour les 3 quartets en fonction du niveau hiérarchique, selon ce système de codage .

Niv	Qrt 1	Qrt 2	Qrt 3	Valeur	Type	Caractéristique
0	1 000	1	//	$81	MF	Unique,sans frères
1	1 001	1	O-F	$91-y	DF	N'ont qu'un parent (MF)
2	1 010	O-F	O-F	SAx-y	DF	Tous les DF
3	1 011	O-F	O-F	SBx-y	DF	nécessitent le
4	1 100	O-F	O-F	SCx-y	DF	3ème quartet
5	1 101	O-F	O-F	SDx-y	DF	pour coder leur
6	1 110	O-F	O-F	SEx-y	DF	numéro dans le
7	1 111	O-F	O-F	SFx-y	DF	niveau

Le contenu des quartets est donné pour le 1er en binaire et pour le 2ème et le 3ème en hexadécimal, sauf pour celui du MF qui n'existe pas. Par convention le 2ème quartet du MF, qui n'a pas de parent, est codé à 1 ce qui entraîne que les 2ème quartets des DF du 1er niveau sont toujours à 1, puisque ces DF n'ont qu'un seul parent, qui est le MF lequel par définition à un numéro 1.

On a placé dans la colonne valeur, selon une numération hexadécimale symbolisée par le signe $, le contenu possible des 3 quartets (2 pour le MF) en partant pour la valeur du MF d'une valeur conventionnelle $81, prévue pour que les fichiers EF du MF n'aient pas, compte tenu de la méthode de codage des

EF expliquée plus loin, une valeur $00, qui pourrait être gênante au niveau du système d'exploitation. Le chiffre x est le numéro du parent et le chiffre y le numéro de création dans le niveau hiérarchique. Ainsi donc on aura la relation, n étant un niveau hiérarchique donné :

$$x_n = y(n-1)$$

Cette convention signifie également que la numérotation du niveau commence à 8 pour le niveau 0, et on constate bien sur le tableau que les niveaux suivants s'étagent de 9 à F en hexadécimal.

A titre d'exemple, un répertoire de valeur $C3-5 est un DF de niveau 4, enfant de Bx-3 et il est le 5ème créé dans ce niveau.

Les fichiers EF sont codés sur 8 bits répartis en 2 quartets :

Le 1er quartet comprend un premier bit (poids fort) qui est toujours à 0 pour indiquer qu'il s'agit d'un EF, et 3 autres bits qui indiquent le niveau hiérarchique où se trouve le EF ainsi codé. Il y a donc bien entendu 8 niveaux hiérarchiques possibles correspondant au MF, dont les fils sont codés par 3 bits 000, et aux 7 niveaux de DF possibles. Le codage de ces 8 niveaux va de 000 à 111.

On constate que ce type de codage correspond au complément à 8 du 1er quartet du DF (ou du MF) dont le EF est l'enfant, ce qui facilite et simplifie le fonctionnement du système d'exploitation.

Le 2ème quartet de l'EF, qui peut prendre les valeurs binaires comprises entre 0000 et 1111, soit de 0 à 15 en décimal ou de 0 à F en hexadécimal, code le numéro du DF dont le EF est l'enfant, afin de pouvoir déterminer la filiation par rapport aux 16 DF possibles dans le niveau hiérarchique où se trouve le EF ainsi

codé. Ceci permet alors au système d'exploitation de remonter, ou de descendre, de proche en proche, l'arborescence qui détermine le fichier EF à utiliser.

On a représenté sur le tableau suivant les contenus possibles des quartets pour les EF, ainsi que leurs valeurs en hexadécimale.

Niv	Qrt A	Qrt B	Valeur	Type	Caractéristique
0	0 000	0001	$01	EF	Tous codés $01
1	0 001	O-F	$10-$1F	EF	
2	0 010	O-F	$20-$2F	EF	
3	0 011	O-F	$30-$3F	EF	
4	0 100	O-F	$40-$4F	EF	
5	0 101	O-F	$50-$5F	EF	
6	0 110	O-F	$60-$6F	EF	
7	0 111	O-F	$70-$7F	EF	

On constate que les EF enfants du MF ont une valeur $01 qui provient justement de la valeur conventionnelle $81 affectée au MF, comme on l'a vu plus haut.

Ainsi par exemple les EF enfants du DF de valeur $C2-5 auront une valeur $4-5.

On a représenté sur la figure 2 la même structure de mémoire que sur la figure 1, mais avec les valeurs de codage des différents répertoires et fichiers. On constate que ceux-ci sont bien identifiables et qu'il est toujours possible de repérer un chemin allant du MF de valeur $81 à un DF quelconque, uniquement à partir des valeurs successives des DF. Le cas de la sélection des fichiers EF sera traité plus loin.

A titre de variante, on peut prévoir d'inverser les

2ème et 3ème quartet de codage des DF. Cette variante pourra éventuellement être utile dans certains systèmes d'exploitations mais elle ne change rien au principe de l'invention tel qu'il a été décrit jusqu'à présent.

On a représenté sur la figure 3 le schéma d'organisation d'une mémoire intégrée dans laquelle sont inscrits les différents répertoires et fichiers, selon le procédé de l'invention décrit ci-dessus.

On suppose que la mémoire est lue séquentiellement, avec éventuellement des sauts possible à une adresse située plus loin.

Le début de la mémoire est bien entendu occupé par une zone système 101 qui est lue lors de l'initialisation du microprocesseur, à la mise sous tension par exemple.

Cette zone système est suivie immédiatement du répertoire principal MF, codé $81, dans une zone 102.

Dans cet exemple, l'application sélectionnée correspond à un DF particulier, par exemple celui ayant la valeur $B2-0, qui se trouve situé dans une zone 104 située en dessous du MF, à une distance variable de celui-ci. Cette distance correspond à un ensemble de répertoires secondaires et de fichiers élémentaires situés dans une zone 103.

En effet, selon l'invention, on place systématiquement les DF et les EF nouvellement créés après les parents dont ils sont issus, mais à une distance quelconque après eux .

Ainsi donc ce DF sélectionné dans la zone 104 est suivi d'une zone 105 comportant elle-même d'autres DF et EF. Cette zone 105 est suivie d'une zone 106 comprenant le 1er EF enfant du DF sélectionné, ce EF ayant la valeur $3-0.

On trouve ensuite une zone quelconque 107, puis une

zone 108 comprenant le 1er DF enfant du DF sélectionné, dont la valeur est donc $C0-0.

Immédiatement après ce 1er DF enfant, on trouve dans une zone 109 le 2ème EF enfant du DF sélectionné. Ce 2ème EF enfant à une valeur $3-0 qui est la même que celle du 1er EF enfant situé dans la zone 106. La distinction se fait parce que cet EF est situé physiquement dans la mémoire après le 1er EF enfant. Il ne peut pas y avoir de confusion quant à la filiation avec les DF, puisqu'en effet le 1er quartet indique le niveau hiérarchique de l'EF enfant, qui est le même que celui du DF dont il est l'enfant, et que le 2ème quartet indique le rang de création du DF parent dans le niveau hiérarchique. Ainsi donc les 2 EF de valeur $3-0 ne peuvent être que les enfants du DF sélectionné de valeur $B2-0 et la distinction entre eux vient de ce que le premier situé dans la zone 106 est au dessus du second situé dans la zone 109.

Le reste de la mémoire, formant une zone 110, pourra contenir, au moins partiellement, des EF et des DF et pourra aussi comporter une partie libre, servant par exemple à stocker les résultats intermédiaires des traitements auxquels a donné accès le DF sélectionné.

L'invention ainsi décrite porte donc sur le système de codage et sur la mise en place dans la mémoire. Elle n'est pas limitée par les chiffres décrits ci-dessus.En particulier on pourra prévoir d'utiliser par exemple 256 répertoires secondaires DF par niveau hiérarchique au lieu de 16. Dans un tel cas il faudra utiliser 4 bits supplémentaires pour coder le numéro du DF dans le niveau hiérarchique, et donc 4 bits supplémentaires pour rappeler le numéro du parent. On aura donc besoin de 20 bits, c'est à dire 5 quartets, pour coder les répertoires secondaires DF. Par conséquent, il faudra 12

bits pour coder les fichiers élémentaires EF.

Le codage selon l'invention occupe donc très peu d'espace, c'est à dire qu'il est très compact. En outre il est très facile à mettre en place et il facilite également la gestion interne du système d'exploitation, ce qui permet de réduire en conséquence la taille du programme de gestion incorporé dans ce système d'exploitation.

REVENDICATIONS

1 - Procédé de reconnaissance des fichiers dans une mémoire, notamment une mémoire pour carte à circuit intégré, du type consistant à prévoir une organisation hiérarchique arborescente du type "parents-enfants" ayant en tête un répertoire principal MF suivi de répertoires secondaires DF et de fichiers élémentaires EF, caractérisé en ce que l'on place dans la mémoire (101-110) le répertoire principal (102) en tête puis les répertoires secondaires (104,108) et les fichiers élémentaires (106,109) à la suite les uns des autres au fur et à mesure de leur création mais en respectant l'ordre hiérarchique, que l'on code le MF et les DF avec un premier mot binaire comprenant un bit indiquant qu'il s'agit du MF ou d'un DF et un ensemble d'autres bits indiquant le niveau hiérarchique du répertoire et un 2ème mot binaire indiquant le numéro du répertoire parent dans son niveau hiérarchique, et pour les DF un 3ème mot binaire permettant de coder le numéro de ce DF à l'intérieur de son niveau hiérarchique, et que l'on code en outre les fichiers élémentaires EF avec un premier mot binaire comportant un bit indiquant qu'il s'agit d'un EF, ce bit étant l'inverse du bit correspondant du MF ou des DF et les autres bits de ce premier mot binaire indiquant le niveau hiérarchique de l'EF, et un deuxième mot binaire identique au troisième mot binaire du DF (ou MF) dont le EF est l'enfant.

2 - Procédé selon la revendication 1, caractérisé en ce que l'on interverti les rôles des 2ème et 3ème mots binaires des DF.

3 - Procédé selon l'une quelconque des

revendications 1 et 2, caractérisé en ce que les mots binaires sont des quartets de 4 bits.

4 - Procédé selon l'une quelconque des revendications 1 à 3, caractérisé en ce que le 1er bit du 1er quartet du MF ou du DF est un 1, que le 1er bit du 1er quartet des EF est un 0, et que le 1er quartet d'un EF est le complément à 8 du 1er quartet du DF dont il est l'enfant.

5 - Procédé selon l'une quelconque des revendications 1 à 4, caractérisé en ce que la valeur hexadécimale du MF est codée à une valeur conventionnelle $81.

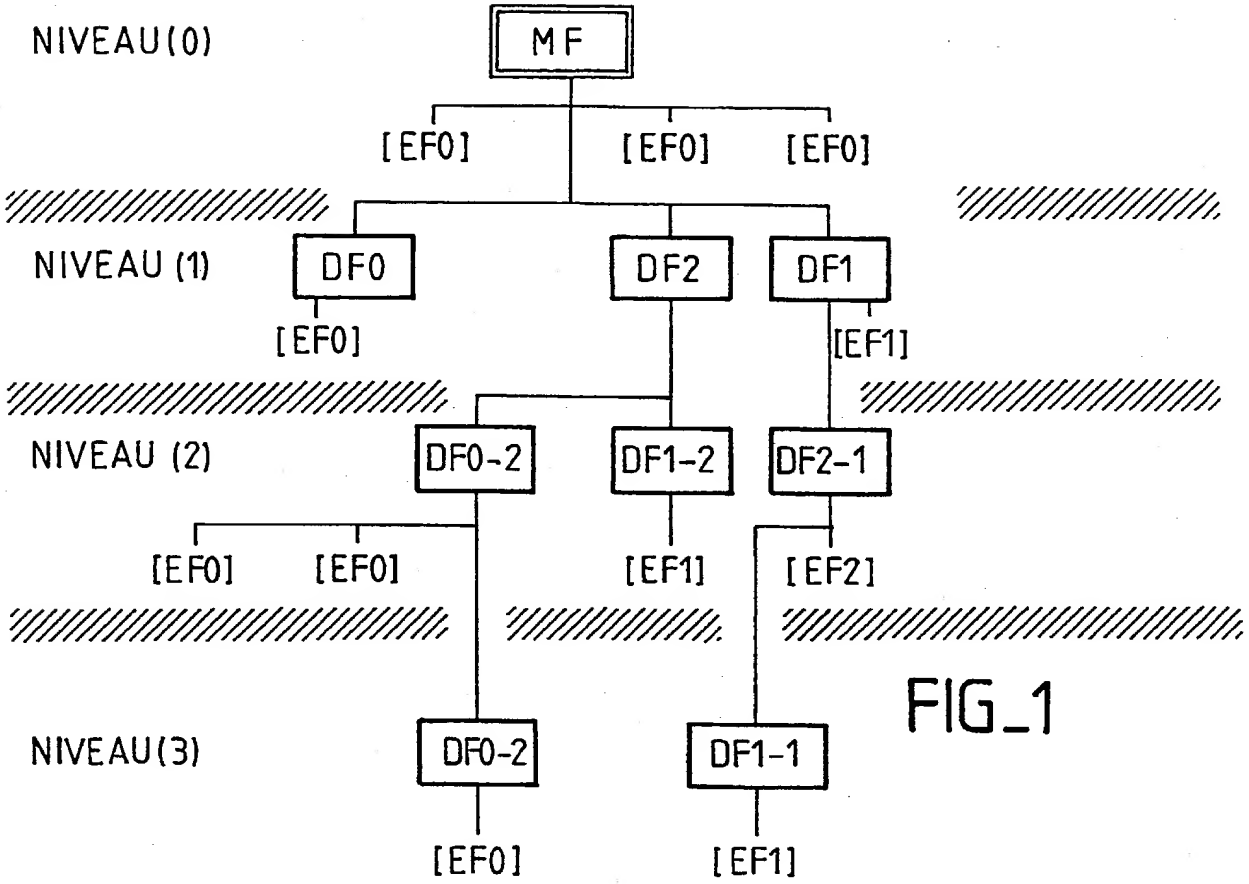

NIVEAU(0)
MF
[EF0]
[EF0]
[EF0]
NIVEAU (1)
DF0
DF2
DF1
[EF0]
[EF1]
NIVEAU (2)
DF0-2
DF1-2
DF2-1
[EF0]
[EF0]
[EF1]
[EF2]
NIVEAU(3)
DF0-2
DF1-1
[EF0]
[EF1]

FIG_1

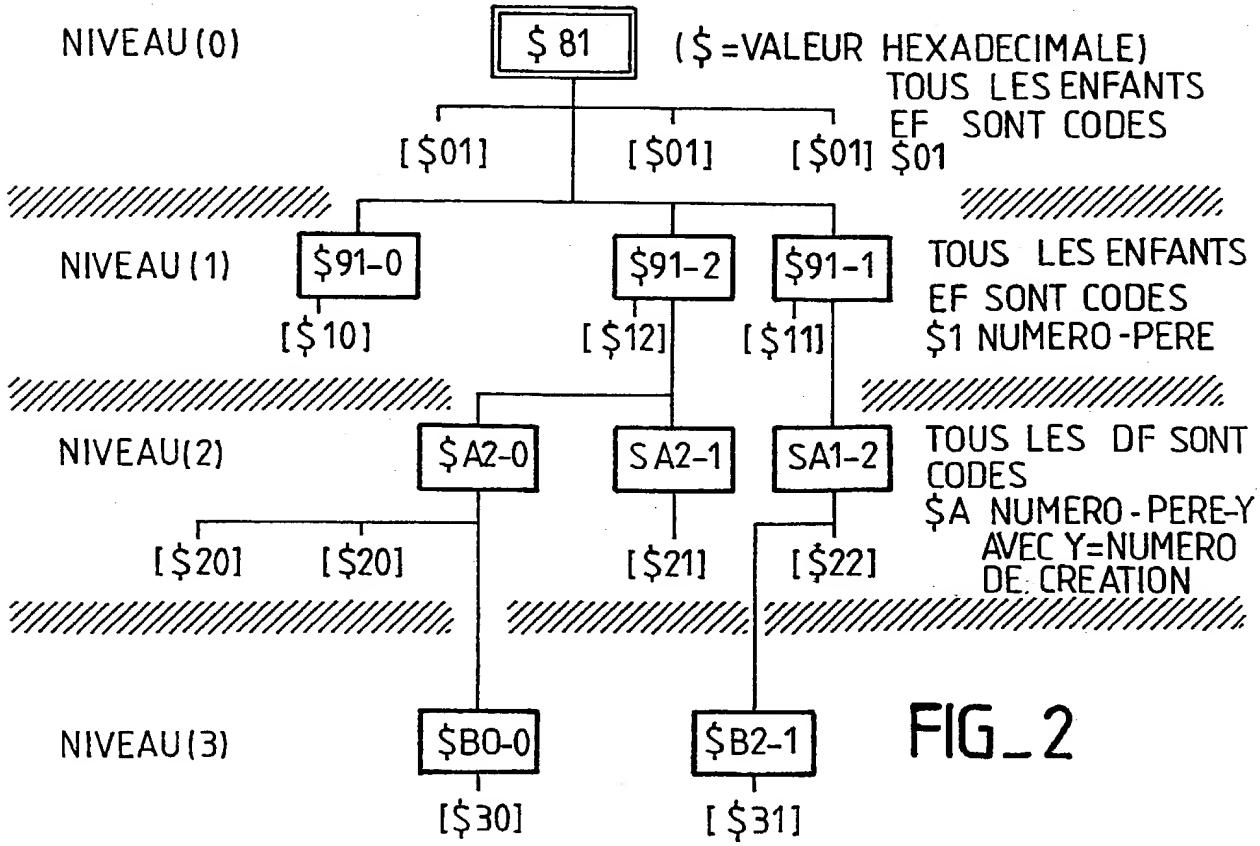
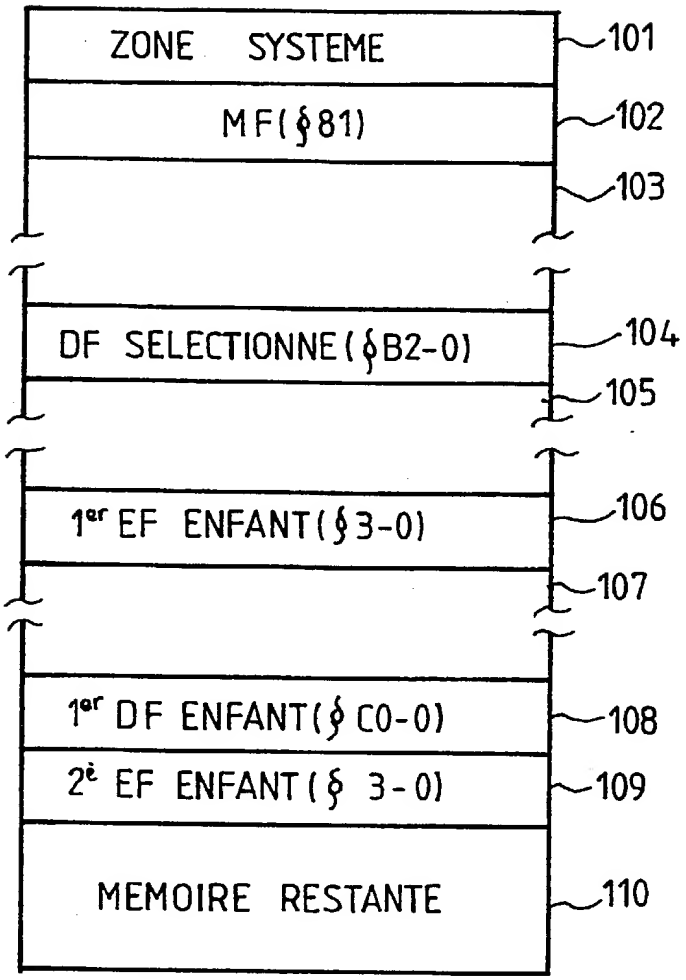

NIVEAU(0)
$ 81
($=VALEUR HEXADECIMALE)
TOUS LES ENFANTS
EF SONT CODES
$01
[$01]
[$01]
[$01]
NIVEAU(1)
$91-0
$91-2
$91-1
TOUS LES ENFANTS
EF SONT CODES
$1 NUMERO-PERE
[$10]
[$12]
[$11]
NIVEAU(2)
$A2-0
SA2-1
SA1-2
TOUS LES DF SONT
CODES
$A NUMERO-PERE-Y
AVEC Y=NUMERO
DE CREATION
[$20]
[$20]
[$21]
[$22]
NIVEAU(3)
$B0-0
$B2-1
[$30]
[$31]

FIG_2

FIG_3


ZONE SYSTEME
101
MF(§81)
102
103
DF SELECTIONNE(§B2-0)
104
105
1er EF ENFANT(§3-0)
106
107
1er DF ENFANT(§C0-0)
108
2è EF ENFANT(§ 3-0)
109
MEMOIRE RESTANTE
110

REPUBLIQUE FRANÇAISE

2693008

INSTITUT NATIONAL
de la
PROPRIETE INDUSTRIELLE

RAPPORT DE RECHERCHE

établi sur la base des dernières revendications déposées avant le commencement de la recherche

N° d'enregistrement national

FR 9208048
FA 473212

DOCUMENTS CONSIDERES COMME PERTINENTS		Revendications concernées de la demande examinée
Catégorie	Citation du document avec indication, en cas de besoin, des parties pertinentes	
A	PROCEEDINGS OF THE ANNUAL INTERNATIONAL CONFERENCE OF THE IEEE ENGINEERING IN MEDICINE AND BIOLOGY SOCIETY vol. 2, 12 Novembre 1989, pages 777 - 778 P. FRENGER MD. 'MCard/FS: A file manager for memory cards' * page 777, colonne 2, ligne 9 - page 777, colonne 2, ligne 19 *	1
A	FR-A-2 611 289 (TOSHIBA K.K.) 26 Août 1988 * abrégé *	1

DOMAINES TECHNIQUES RECHERCHES (Int. Cl.5)

G06F
G06K

Date d'achèvement de la recherche
25 FEVRIER 1993

Examinateur
KATERBAU R.E.

CATEGORIE DES DOCUMENTS CITES

X : particulièrement pertinent à lui seul
Y : particulièrement pertinent en combinaison avec un autre document de la même catégorie
A : pertinent à l'encontre d'au moins une revendication ou arrière-plan technologique général
O : divulgation non-écrite
P : document intercalaire

T : théorie ou principe à la base de l'invention
E : document de brevet bénéficiant d'une date antérieure à la date de dépôt et qui n'a été publié qu'à cette date de dépôt ou qu'à une date postérieure.
D : cité dans la demande
L : cité pour d'autres raisons

& : membre de la même famille, document correspondant

EPO FORM 1503 03.82 (P0413)

1